| Ref # | Hits | Search Query | DBs | Default Operator | Plurals | Time Stamp |
|---|---|---|---|---|---|---|
| L1 | 7 | target adj design adj model | US-PGPUB;<br>USPAT;<br>EPO;<br>DERWENT | OR | OFF | 2005/09/14 11:48 |
| L2 | 0 | ("6691078").URPN. | USPAT | OR | OFF | 2005/09/14 11:51 |
| L3 | 0 | ("6691078").URPN. | USPAT | OR | OFF | 2005/09/14 11:51 |
| L4 | 14 | ("5394347" \| "5615137" \| "5910897" \| "5926622" \| "5946481" \| "5966516" \| "6044211" \| "6131078" \| "6163876" \| "6178394" \| "6185516" \| "6275976" \| "6311293" \| "6353896").PN. | US-PGPUB;<br>USPAT;<br>USOCR | OR | OFF | 2005/09/14 11:52 |

| Ref # | Hits | Search Query | DBs | Default Operator | Plurals | Time Stamp |
|---|---|---|---|---|---|---|
| L4 | 14 | ("5394347" \| "5615137" \| "5910897" \| "5926622" \| "5946481" \| "5966516" \| "6044211" \| "6131078" \| "6163876" \| "6178394" \| "6185516" \| "6275976" \| "6311293" \| "6353896").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2005/09/14 17:20 |
| L5 | 1057 | 714/25.ccls. | US-PGPUB; USPAT; EPO; DERWENT | OR | OFF | 2005/09/14 17:20 |
| L6 | 678 | L5 and @ad<"20010112" | US-PGPUB; USPAT; EPO; DERWENT | OR | OFF | 2005/09/14 17:21 |
| L7 | 2 | L6 and model adj check$3 | US-PGPUB; USPAT; EPO; DERWENT | OR | OFF | 2005/09/14 17:22 |
| L8 | 418 | 714/45.ccls. | US-PGPUB; USPAT; EPO; DERWENT | OR | OFF | 2005/09/14 17:22 |
| L9 | 307 | L8 and @ad<"20010112" | US-PGPUB; USPAT; EPO; DERWENT | OR | OFF | 2005/09/14 17:22 |
| L10 | 58 | L9 and model | US-PGPUB; USPAT; EPO; DERWENT | OR | OFF | 2005/09/14 17:28 |
| L11 | 0 | patrh adj expression$1 | US-PGPUB; USPAT; EPO; DERWENT | OR | OFF | 2005/09/14 17:28 |
| L12 | 252 | path adj expression$1 | US-PGPUB; USPAT; EPO; DERWENT | OR | OFF | 2005/09/14 17:29 |
| L13 | 101 | L12 and @ad<"20010113" | US-PGPUB; USPAT; EPO; DERWENT | OR | OFF | 2005/09/14 17:29 |

CiteSeer Find: path expressions and model Documents Citations

Searching for **path expressions and model**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP
40 documents found. **Order: number of citations.**

A Reflective Approach to Process Model Customization.. - Jamart, van Lamsweerde (1994) (Correct) (4 citations)
acting on objects 4]Some kind of **path expressions** and production rules guarded by events may
A Reflective Approach to Process **Model** Customization, Enactment and Evolution Philippe
processes are enacted instances of process **models**. Process **models** are themselves instances of
ftp.info.ucl.ac.be/pub/publi/94/ICSP3.ps

The Role of Polymorphic Reuse Mechanisms in Schema Evolution .. - Liu, Zicari, al. (1997) (Correct) (4 citations)
or queries must be associated with a precise **path expression** in order to traverse the nested structure of
as an interesting specification formalism for **model**ing operational requirements. They encourage the
seen as an interesting specification formalism for **model**ing operational requirements in object-oriented
web.cs.ualberta.ca/~lingliu/evolve/tkde.ps

Implementation Aspects of an Object-Oriented DBMS - Dogac (1995) (Correct) (2 citations)
and search-expression may include **path expressions** and methods. MOOD allows dynamic schema
engine. A detailed description of the MOOD data **model** and MOODSQL is given in [Dog 94a, Dog 94c, Ozk 94a, Dog 94c, Ozk 93a, Ozk 93b]In the MOOD data **model**, the basic data Figure 2. Display of the
ftp.srdc.metu.edu.tr/pub/mood/papers/sigmod.ps.Z

A Heuristic Approach for Optimization of Path Expressions - Cetin Ozkan (1995) (Correct) (2 citations)
A Heuristic Approach for Optimization of **Path Expressions** Cetin Ozkan, Asuman Dogac and Cem
common subexpressions. In section 2, the cost **model** is presented and the formulas for the selectivity
Finally, Section 6 contains the conclusions. 2 Cost **Model** In the object **model** [Atk 92] used in this paper,
ftp.srdc.metu.edu.tr/pub/mood/papers/heuristic_dexa_95.ps.Z

How to Write F-Logic Programs in FLORID - A.. - Frohn.. (1997) (Correct) (1 citation)
.17 7 **Path Expressions** 19 7.1 Nesting of **Path Expressions** and
ftp.informatik.uni-freiburg.de/pub/florid/tutorial.ps.gz

A Join Algorithm Utilizing Multiple Path Indexes in.. - Cho, Lee, Yoon, Whang (1996) (Correct) (1 citation)
various types of path indexes -indexes on **path expressions** -have been proposed for efficient
In this paper, we use the object-oriented query **model** proposed by Kifer et al.11]An important
et al.11]An important feature of Kifer's query **model** is the extended **path expression**: a A pred1 B
dblab.kaist.ac.kr/Publication/Ps/eccs96c.ps

Query Processing in Object-Oriented Database Systems - Özsu, Blakeley (1994) (Correct) (1 citation)
Accessing such complex objects involves **path expressions**. The optimization of **path expressions** is a
The second is the optimization of **path expressions**. **Path expressions** represent traversal paths
has seen significant research in defining query **models** (including calculi, algebra and user languages)
web.cs.ualberta.ca/~database/publications/ozsu/wonkim/kimchap.ps.Z

A Java Application Programming Interface to a Multimedia.. - Boll, Wäsch (1996) (Correct) (1 citation)
method calls, of complex data types, and of **path expressions** in VQL queries. 2.3 VODAK transaction
for integrated DBMS support, i.e.the integrated **model**ing, management, and interactive presentation of
DBMSs (OODBMS) form a suitable basis for **model**ing, storage, and management of different media
ftp.darmstadt.gmd.de/pub/oasys/reports/P-96-24.ps.Z

FLORID: A DOOD-System for Querying the Web - Himmeröder, Lausen, Ludäscher.. (1998) (Correct) (1 citation)
classes. Moreover, FLORID extends F-logic by **path expressions** [FLU94] which turn out to be particularly
while object-oriented features allow a flexible **model**ing of semistructured and partial information. 2.
languages for wrappers and mediators. 3. The Web **Model** of FLORID distinguishes between the class url of
www.informatik.uni-freiburg.de/~dbis/Publications/98/edbt98-florid-demo.ps.gz

Genetic Algorithms for Optimal Logical Database Design - van Bommel, Lucasius, van.. (1994) (Correct)

(1 citation)
allow an elegant verbalization of queries by **path expressions**, built from names of object types and role
viz. the transformation of conceptual data **models** into efficient internal representations. Although
process of information analysis. Independent of the **modell**ing technique used (OO, ER, NIAM)the
ftp.cs.kun.nl/pub/SoftwEng.InfSyst/articles/GenAlgDbOpt.ps.Z

Using Graph Transformation Techniques for Integrating.. - Faulstich (1998) (Correct)
data by using for instance regular **path expressions** or automatic type coercion. Examples of
the access to it. The HyperView methodology **models** Web Sites and their HTML pages as graphs from
of views. We present the clustered graph data **model** (CGDM) used in the HyperView system. Views are
www.inf.fu-berlin.de/~faulstic/pub/tagt98.ps.gz

Query Modification in Object-Oriented Database Federations - Vermeer, Apers (Correct)
features such as queries involving **path expressions** and nesting. We show how properties of the
nesting. We show how properties of the OO-style of **modell**ing relationships through object references can
the current popularity of the object-oriented **model** as a canonical data **model** in such federations
wwwis.cs.utwente.nl:8080/isdoc/confpaper/vermeer.coopis97.accepted.ps.gz

Path Constraints from a Modal Logic Point of View - Alechina, de Rijke (Correct)
and Kleene star .They are referred to as **path expressions**. Firstorder logic extended with the
It is generally agreed that the appropriate data **model** for semistructured data is an edge-labelled
2 introduces some background information on data **models**, query languages and logical formalisms used
www.cs.nott.ac.uk/~nza/paths.ps.gz

The Decidability of Some Restricted Implication Problems.. - Buneman, Fan, Weinstein (1997) (Correct)
the optimization techniques for generalized **path expressions** in object-oriented databases developed by
path inclusion constraints, P ,for the graph data **model**. These path constraints are capable of expressing
we first recall the definitions of the data **model** and the constraint language P from [10]and then
ftp.cis.upenn.edu/pub/papers/db-research/tr9715.ps.gz

Type Inference for Queries on Semistructured Data (Extended.. - Milo, Suciu (1999) (Correct)
for semistructured data, allowing regular **path expressions** and joins (both reviewed in Section 2)A
of their attributes stored with the object. The **model** has proven successful in a number of applications
DTD's, and these query languages. 2 Background Data **Model** Our semistructured data **model** is an ordered
ftp.math.tau.ac.il/pub/milo/pods99.ps.Z

FLORID - User Manual Version 2.0 - Frohn, Himmeröder, Kandzia.. (1997) (Correct)
non-monotonic inheritance and furthermore **path expressions** [6] which can also be used for anonymous
of the structural aspects of object-oriented data **model**ing. Contrasting other approaches, e.g. 8]GULOG
To learn something about the language and data **model**ing with F-logic the reader is referred to the
ftp.informatik.uni-freiburg.de/pub/florid/manual.ps.gz

Chase and Axioms for PC Queries and Dependencies - Popa, Tannen (1998) (Correct)
rest of the paper. It describes well-defined **path expressions**, well-defined tableaux, and gives the
v :x !I that can be extended to **path expressions** and path conjunctions over ~x (i.e. v(R) R I
dependencies and equivalences beyond the relational **model** on which we use the equational chase defined
www.cis.upenn.edu/~val/tr-MS-CIS-98-34.ps

Multimedia User Interfaces - Götze, Boles, Eirund (Correct)
of multimedia objects is specified by **path expressions** including path operators which define the
media. Discrete media imply a transitional **model** of dialog control. Each event caused by an user
the integration of continuous media causes this **model** to become inadequate because actions like
www-is.offis.uni-oldenburg.de/~dibo/paper/../paper/mm-userinterfaces.ps.gz

Vql : A Visual Query Language For Uniform Database Access - Yuksel Alp (Correct)
User Friendliness Considerations 95 15.1 Long **Path Expressions** :
It facilitates querying databases of several **models** in a uniform and consistent way. We demonstrate
2 2 VQL Through Examples -The Relational **Model** 5 2.1 Safety :
www.eecs.uic.edu/~yaslando/mt.ps.Z

Grammar Based Information Modelling - Hofstede, Proper, van der Weide (1994) (Correct)
semantics of LISA-D is expressed in terms of **path expressions**, a (for this purpose) convenient variant of
in [HPW93] by means of a translation to **path expressions**. **Path expressions** form the semantical base for
Grammar Based Information **Modell**ing A.H.M. ter Hofstede 1 and H.A. Proper 23
ftp.cs.kun.nl/pub/SoftwEng.InfSyst/articles/ORMGrammar.ps.Z

*Documents 21 to 40* Previous 20

Try your query at: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP

Google Scholar BETA

model checking | 1951 - 2001 | Search | Adv Sch Sch

**Scholar** Results **1** - **10** of about **187,000** for **model checking**. (**0.11** seconds)

[PS] **Model Checking**
EM Clarke, O Grumberg, D Peled - 1999 - pst.ifi.lmu.de
Chapter 21 **Model checking** Edmund M. Clarke Bernd-Holger Schlinglo Contents 1
Introduction . . . . . ... 1436 8 Basic **Model Checking** Algorithms . . . . . ...
Cited by 1931 - View as HTML - Web Search - pst.informatik.uni-muenchen.de - informatik.hu-berlin.de - staff.science.uva.nl - all 19 versions »

[BOOK] Symbolic **Model Checking**
KL McMillan - 1993 - Kluwer Academic Publishers, Norwell, MA, USA
Cited by 1528 - Web Search - Library Search

Symbolic **Model Checking**: 10^20 States and Beyond
JR Burch, EM Clarke, KL McMillan, DL Dill, LJ ... - Information and Computation, 1992 - ieeexplore.ieee.org
Page 1 JR Burch KL McMillan Symbolic **Model Checking**: 1020 States and Beyond
CH2897-7/90/0000/0428$01 .00 © 1990 IEEE 4214 EM Clarke School of Computer Science ...
Cited by 1273 - Web Search - heat.vlsilab.iecs.fcu.edu.tw - portal.acm.org - www-cse.ucsd.edu - all 9 versions »

**Model-checking** for real-time systems
R Alur, C Courcoubetis, D Dill - Logic in Computer Science, 1990. LICS'90, Proceedings., ..., 1990 - ieeexplore.ieee.org
... As the main result, an algorithm is developed for **model checking**, that is, for determining the truth of a TCTL formula with respect to a timed graph. ...
Cited by 404 - Web Search

Symbolic **Model Checking** without BDDs
A Biere, A Cimatti, EM Clarke, Y Zhu - TACAS, 1999 - springerlink.com
Page 1. Symbolic **Model Checking** without BDDs ... Symbolic **Model Checking** [3, 14] has proven to be a powerful tech- nique for the verification of reactive systems. ...
Cited by 502 - Web Search - ra.informatik.uni-stuttgart.de - reports-archive.adm.cs.cmu.edu - people.inf.ethz.ch - all 27 versions »

Symbolic **model checking** for real-time systems
TA Henzinger, X Nicollin, J Sifakis, S Yovine, MZL ... - 1992 - ieeexplore.ieee.org
Page 1 Symbolic **Model Checking** for Real-time Systems* O8l86.2735.22 $3.00 © 1992
IEEE 394 Thomas A. Henzingert Computer Science Department, Corne1 University ...
Cited by 568 - Web Search - www-cad.eecs.berkeley.edu - mtc.epfl.ch - portal.acm.org - all 10 versions »

**Model Checking** and Abstraction
EM Clarke, O Grumberg, DE Long - ACM Transactions on Programming Languages and Systems, 1994 - portal.acm.org
Page 1. **Model Checking** and Abstraction EDMUND M. CLARKE Carnegie Mellon University
ORNA GRUMBERG ... BDDs), **model checking**, temporal loglc 1 ...
Cited by 551 - Web Search - www-2.cs.cmu.edu - engineering.usu.edu - spy.sci.univr.it - all 11 versions »

[PS] PVS: Combining Specification, Proof **Checking**, and **Model Checking**
S Owre, S Rajan, JM Rushby, N Shankar, MK Srivas, ... - CAV, 1996 - casl.csa.iisc.ernet.in
... and **Model Checking** ? ... **Model checking** is one such decision procedure that is used to discharge temporal properties of specic nite-state systems. ...
Cited by 267 - View as HTML - Web Search - informatik.uni-freiburg.de - people.cs.uchicago.edu - csl.sri.com - all 10 versions »

[CITATION] Symbolic **model checking**: an approach to the state explosion problem

KL McMillan - 1992 - Pittsburgh, Pa.: School of Computer Science, Carnegie Mellon ...
Symbolic **model checking** : an approach to the state explosion problem. By: K L McMillan.
Type: English : Book : Non-fiction. Publisher: Pittsburgh, Pa. **...**
Cited by 532 - Web Search - portal.acm.org - Library Search

**Model-Checking** in Dense Real-time
R Alur, C Courcoubetis, DL Dill - Information and Computation, 1993 - portal.acm.org
**...** Search: The ACM Digital Library The Guide. Feedback Report a problem
Satisfaction survey. **Model-checking** in dense real-time. Source, **...**
Cited by 382 - Web Search - cis.upenn.edu - cm.bell-labs.com - all 4 versions »

Gooooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

model checking Search

Google Home - About Google - About Google Scholar

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ○ The ACM Digital Library ● The Guide

model checking SEARCH

THE GUIDE TO COMPUTING LITERATURE

Feedback Report a problem Satisfaction survey

and Published before February 2001
Terms used **model checking**

Found **1,742** of **1,742**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The Digital Library

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale □ ▂ ▄ ▆ ■

1 An Efficient Algorithm for Real-Time Symbolic Model Checking ■
J. Froessl, Th. Kropf, J. Gerlach
March 1996 **Proceedings of the 1996 European conference on Design and Test**

Full text available: pdf(813.38 KB) Publisher Site Additional Information: full citation, abstract, citings

The verification of real-time properties requires model checking techniques for quantitative temporal structures and real-time temporal logics. However, up to now, most of those problems were solved by a translation into a standard CTL model checking problem with unit-delay structures. Although usual CTL model checkers like SMV can be used then, the translation leads to large structures and CTL formulas, such that the verification requires large computation times and only small circuits can be v ...

**Keywords**: formal verification, model-checking, real-time systems, quantitative temporal logic

2 Another Look at LTL Model Checking ■
Edmund M. Clarke, Orna Grumberg, Kiyoharu Hamaguchi
February 1997 **Formal Methods in System Design**, Volume 10 Issue 1

Full text available: Publisher Site Additional Information: full citation, abstract, references, citings

We show how LTL model checking can be reduced to CTL model checking with fairness constraints. Using this reduction, we also describe how to construct a {\em symbolic} LTL model checker that appears to be quite efficient in practice. In particular, we show how the SMV model checking system developed by McMillan [16] can be extended to permit LTL specifications. The results that we have obtained are quite surprising. For the specifications wh ...

**Keywords**: automatic verification, binary decision diagrams, model checking, temporal logic

3 Refining Model Checking by Abstract Interpretation ■
Patrick Cousot, Radhia Cousot
January 1999 **Automated Software Engineering**, Volume 6 Issue 1

Full text available: Publisher Site Additional Information: full citation, abstract, citings

Formal methods combining abstract interpretation and model-checking have been considered for automated analysis of software.

In abstract model-checking, the semantics of an infinite transition system is abstracted to get a finite approximation on which temporal-logic/μ-calculus model-checking can be directly applied.

The paper proposes two improvements of abstract model-checking which can be applied to infinite abstract transition systems:

iA new combination of fo ...

**Keywords**: abstract interpretation, model-checking, static analysis, transition system, universal safety

4 Model Checking Complete Requirements Specifications Using Abstraction

Ramesh Bharadwaj, Constance L. Heitmeyer

January 1999 **Automated Software Engineering**, Volume 6 Issue 1

Full text available: Publisher Site Additional Information: full citation, abstract, citings

Although model checking has proven remarkably effective in detecting errors in hardware designs, its success in the analysis of software specifications has been limited. Model checking algorithms for hardware verification commonly use Binary Decision Diagrams (BDDs) to represent predicates involving the many Boolean variables commonly found in hardware descriptions. Unfortunately, BDD representations may be less effective for analyzing software specifications, which usually contain no ...

**Keywords**: SCR, abstraction, model checking, requirements specification, verification

5 Selective Quantitative Analysis and Interval Model Checking: Verifying Different Facets of a System

Sérgio Campos, Edmund M. Clarke, Orna Grumberg

October 2000 **Formal Methods in System Design**, Volume 17 Issue 2

Full text available: Publisher Site Additional Information: full citation, abstract, references, index terms

In this work we propose a verification methodology consisting of *selective quantitative timing analysis* and *interval model checking*. Our methods can aid not only in determining if a system works correctly, but also in understanding how *well* the system works. The selective quantitative algorithms compute minimum and maximum delays over a selected subset of system executions. A linear-time temporal logic (LTL) formula is used to select either infinite paths or finite interv ...

**Keywords**: LTL model checking, quantitative timing analysis, real-time systems, symbolic model checking, temporal logic model checking

6 Using predicate abstraction to reduce object-oriented programs for model checking

William Visser, SeungJoon Park, John Penix

August 2000 **Proceedings of the third workshop on Formal methods in software practice**

Full text available: pdf(385.20 KB) Additional Information: full citation, abstract, references, citings, index terms

While it is becoming more common to see model checking applied to software requirements specifications, it is seldom applied to software implementations. The Automated Software Engineering group at NASA Ames is currently investigating the use of model checking for actual source code, with the eventual goal of allowing software developers to augment traditional testing with model checking. Because model checking suffers from the state-explosion problem, one of the main hurdles for program ...

**Keywords**: browsing

7 An automata-theoretic approach to modular model checking
Orna Kupferman, Moshe Y. Vardi
January 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 1

Full text available: pdf(458.27 KB) Additional Information: full citation, abstract, references, citings, index terms

In modular verification the specification of a module consists of two part. One part describes the guaranteed behavior of the module. The other part describes the assumed behavior of the system in which the module is interacting. This is called the assume-guarantee paradigm. In this paper we consider assume-guarantee specifications in which the guarantee is specified by branching temporal formulas. We distinguish between two approaches. In the first approac ...

**Keywords**: automata, modular verification, temporal logic

8 Symbolic guided search for CTL model checking
Roderick Bloem, Kavita Ravi, Fabio Somenzi
June 2000 **Proceedings of the 37th conference on Design automation**

Full text available: pdf(110.11 KB) Additional Information: full citation, abstract, references, citings, index terms

CTL model checking of complex systems often suffers from the state-explosion problem. We propose using Symbolic Guided Search to avoid difficult-to-represent sections of the state space and prevent state explosion from occurring. Symbolic Guided Search applies hints to guide the exploration of the state space. In this way, the size of the BDDs involved in the computation is controlled, and the truth of a property may be decided before all states have been explored. In this work, ...

9 Action Language: a specification language for model checking reactive systems
Tevfik Bultan
June 2000 **Proceedings of the 22nd international conference on Software engineering**

Full text available: pdf(299.62 KB) Additional Information: full citation, abstract, references, citings, index terms

We present a specification language called Action Language for model checking software specifications. Action Language forms an interface between transition system models that a model checker generates and high level specification languages such as Statecharts, RSML and SCR—similar to an assembly language between a microprocessor and a programming language. We show that Action Language translations of Statecharts and SCR specifications are compact and they preserve the structure of th ...

**Keywords**: model checking, reactive systems, specification languages

10 Composite model-checking: verification with type-specific symbolic representations
Tevfik Bultan, Richard Gerber, Christopher League
January 2000 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 9 Issue 1

Full text available: pdf(400.17 KB) Additional Information: full citation, abstract, references, citings, index terms

There has been a surge of progress in automated verification methods based on state exploration. In areas like hardware design, these technologies are rapidly augmenting key phases of testing and validation. To date, one of the most successful of these methods has been symbolic model-checking, in which large finite-state machines are encoded into compact data structures such as Binary Decision Diagrams (BDDs), and are then checked for safety and liveness properties. However, these technique ...

**Keywords**: Presburger arithmetic, binary decision diagrams, symbolic model-checking

11 An automata-theoretic approach to branching-time model checking
Orna Kupferman, Moshe Y. Vardi, Pierre Wolper
March 2000 **Journal of the ACM (JACM)**, Volume 47 Issue 2

Full text available: pdf(379.70 KB) Additional Information: full citation, abstract, references, citings, index terms

Translating linear temporal logic formulas to automata has proven to be an effective approach for implementing linear-time model-checking, and for obtaining many extensions and improvements to this verification method. On the other hand, for branching temporal logic, automata-theoretic techniques have long been thought to introduce an exponential penalty, making them essentially useless for model-checking. Recently, Bernholtz and Grumberg [1993] have shown that this exponential penalty can ...

12 Deductive Model Checking
Henny B. Sipma, Tomás E. Uribe, Zohar Manna
July 1999 **Formal Methods in System Design**, Volume 15 Issue 1

Full text available: Publisher Site Additional Information: full citation, abstract, references, citings, index terms

We present an extension of classical tableau-based model checking procedures to the case of infinite-state systems, using deductive methods in an incremental construction of the behavior graph. Logical formulas are used to represent infinite sets of states in an abstraction of this graph, which is repeatedly refined in the search for a counterexample computation, ruling out large portions of the graph before they are expanded to the state-level. This can lead to large savings, even i ...

**Keywords**: deductive verification, model checking, reactive systems

13 Model-checking concurrent systems with unbounded integer variables: symbolic representations, approximations, and experimental results
Tevfik Bultan, Richard Gerber, William Pugh
July 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 21 Issue 4

Full text available: pdf(467.20 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Model checking is a powerful technique for analyzing large, finite-state systems. In an infinite state system, however, many basic properties are undecidable. In this article, we present a new symbolic model checker which conservatively evaluates safety and liveness properties on programs with unbounded integer variables. We use Presburger formulas to symbolically encode a program's transition system, as well as its model-checking computations. All fixpoint calculations are executed symbol ...

**Keywords**: Presburger arithmetic, abstract interpretation, symbolic model checking

14 Modalities for model checking (extended abstract): branching time strikes back
E. Allen Emerson, Chin-Laung Lei
January 1985 **Proceedings of the 12th ACM SIGACT-SIGPLAN symposium on Principles of programming languages**

Full text available: pdf(1.19 MB) Additional Information: full citation, citings, index terms

15 Decoupling synchronization from local control for efficient symbolic model checking of statecharts
William Chan, Richard J. Anderson, Paul Beame, David H. Jones, David Notkin, William E. Warner
May 1999 **Proceedings of the 21st international conference on Software engineering**
Full text available: pdf(1.31 MB) Additional Information: full citation, references, citings, index terms

**Keywords**: binary decision diagrams, fault tolerance, formal methods, formal verification, software specification, statecharts, symbolic model checking

16 Approximate reachability don't cares for CTL model checking
In-Ho Moon, Jae-Young Jang, Gary D. Hachtel, Fabio Somenzi, Jun Yuan, Carl Pixley
November 1998 **Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design**
Full text available: pdf(938.98 KB) Additional Information: full citation, references, citings, index terms

17 Filter-based model checking of partial systems
Matthew B. Dwyer, Corina S. Pasareanu
November 1998 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering**, Volume 23 Issue 6
Full text available: pdf(1.43 MB) Additional Information: full citation, abstract, references, citings, index terms

Recent years have seen dramatic growth in the application of model checking techniques to the validation and verification of correctness properties of hardware, and more recently software, systems. Most of this work has been aimed at reasoning about properties of complete systems. This paper describes an automatable approach for building finite-state models of partially defined software systems that are amenable to model checking using existing tools. It enables the application of existing model ...

**Keywords**: assume-guarantee reasoning, filter-based analysis, model checking, software verification and validation

18 Model checking of hierarchical state machines
Rajeev Alur, Mihalis Yannakakis
November 1998 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering**, Volume 23 Issue 6
Full text available: pdf(1.09 MB) Additional Information: full citation, abstract, references, citings, index terms

Model checking is emerging as a practical tool for detecting logical errors in early stages of system design. We investigate the model checking of hierarchical (nested) systems, i.e. finite state machines whose states themselves can be other machines. This nesting ability is common in various software design methodologies and is available in several commercial modeling tools. The straightforward way to analyze a hierarchical machine is to flatten it (thus, incurring an exponential blow up) and a ...

19 Using "test model-checking" to verify the Runway-PA8000 memory model
Rajnish Ghughal, Abdel Mokkedem, Ratan Nalumasu, Ganesh Gopalakrishnan
June 1998 **Proceedings of the tenth annual ACM symposium on Parallel algorithms and architectures**
Full text available: pdf(1.14 MB) Additional Information: full citation, references, citings, index terms

20 Isomorph-free model enumeration: a new method for checking relational specifications

Daniel Jackson, Somesh Jha, Craig A. Damon

March 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 2

Full text available: pdf(273.78 KB) Additional Information: full citation, abstract, references, citings, index terms

Software specifications often involve data structures with huge numbers of value, and consequently they cannot be checked using standard state exploration or model-checking techniques. Data structures can be expressed with binary relations, and operations over such structures can be expressed as formulae involving relational variables. Checking properties such as preservation of an invariant thus reduces to determining the validity of a formula or, equivalently, finding a model (of the form ...

**Keywords**: formal specification, model checking, model finding, object models, pruning, relational calculus, relational specifications, symmetry

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Results for "**((model checking<and>path expression)) <and> (pyr >= 1951 <and> pyr <= 2001)**" e-mail
Your search matched **1** of **1235066** documents.
A maximum of **500** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

**» Search Options**

View Session History

New Search

**Modify Search**

((model checking<and>path expression)) <and> (pyr >= 1951 <and> pyr <= 2001) »

□ Check to search only within this results set

**Display Format:** ● Citation ○ Citation & Abstract

**» Key**

| | |
|---|---|
| **IEEE JNL** | IEEE Journal or Magazine |
| **IEE JNL** | IEE Journal or Magazine |
| **IEEE CNF** | IEEE Conference Proceeding |
| **IEE CNF** | IEE Conference Proceeding |
| **IEEE STD** | IEEE Standard |

□ 1. **Using paths to measure, explain, and enhance program behavior**
Ball, T.; Larus, J.R.;
Computer
Volume 33, Issue 7, July 2000 Page(s):57 - 65
Digital Object Identifier 10.1109/2.869371
AbstractPlus | References | Full Text: PDF(144 KB) **IEEE JNL**

[redacted]

Help Contact Us Privacy & :

IEEE Xplore® RELEASE 2.1

Welcome United States Patent and Trademark Office

□ **Search Results**

Results for "**((model checking<and>path expressions)) <and> (pyr >= 1951 <and> pyr <= 2001...**"

Your search matched **1** of **1235066** documents.

A maximum of **500** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

e-mail

**» Search Options**

View Session History

New Search

**Modify Search**

((model checking<and>path expressions)) <and> (pyr >= 1951 <and> pyr <= 2001) »

□ Check to search only within this results set

**Display Format:** ● Citation ○ Citation & Abstract

**» Key**

| | |
|---|---|
| **IEEE JNL** | IEEE Journal or Magazine |
| **IEE JNL** | IEE Journal or Magazine |
| **IEEE CNF** | IEEE Conference Proceeding |
| **IEE CNF** | IEE Conference Proceeding |
| **IEEE STD** | IEEE Standard |

□ 1. **Using paths to measure, explain, and enhance program behavior**
Ball, T.; Larus, J.R.;
Computer
Volume 33, Issue 7, July 2000 Page(s):57 - 65
Digital Object Identifier 10.1109/2.869371
AbstractPlus | References | Full Text: PDF(144 KB) **IEEE JNL**

[redacted]

Help Contact Us Privacy &

Indexed by Inspec®

**IEEE Xplore** RELEASE 2.1

Home | Login | Logout | Access Information | Alerts |

**Welcome United States Patent and Trademark Office**

□ **Search Results**

**BROWSE** **SEARCH** **IEEE XPLORE GUIDE**

Results for "**(path expressions) <and> (pyr >= 1951 <and> pyr <= 2001)**"
Your search matched **91** of **1235066** documents.
A maximum of **500** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

e-mail

**» Search Options**

View Session History

New Search

**» Key**

| | |
|---|---|
| IEEE JNL | IEEE Journal or Magazine |
| IEE JNL | IEE Journal or Magazine |
| IEEE CNF | IEEE Conference Proceeding |
| IEE CNF | IEE Conference Proceeding |
| IEEE STD | IEEE Standard |

**Modify Search**

(path expressions) <and> (pyr >= 1951 <and> pyr <= 2001) »

□ Check to search only within this results set

**Display Format:** ● Citation ○ Citation & Abstract

**Select** **Article Information** **View: 1-25** | 26-

□ 1. **Forward vs. reverse traversal in path expression query processing**
Taniar, D.;
Technology of Object-Oriented Languages, 1998. TOOLS 28. Proceedings
23-26 Nov. 1998 Page(s):127 - 140
Digital Object Identifier 10.1109/TOOLS.1998.750030
AbstractPlus | Full Text: PDF(584 KB) **IEEE CNF**

□ 2. **Ensuring critical event sequences in high consequence computer based inspired by path expressions**
Kidd, M.-E.C.;
Engineering of Computer-Based Systems, 1997. Proceedings., International C Workshop on
24-28 March 1997 Page(s):483 - 490
Digital Object Identifier 10.1109/ECBS.1997.581943
AbstractPlus | Full Text: PDF(704 KB) **IEEE CNF**

□ 3. **Representative objects: concise representations of semistructured, hiera**
Nestorov, S.; Ullman, J.; Wiener, J.; Chawathe, S.;
Data Engineering, 1997. Proceedings. 13th International Conference on
7-11 April 1997 Page(s):79 - 90
Digital Object Identifier 10.1109/ICDE.1997.581741
AbstractPlus | Full Text: PDF(976 KB) **IEEE CNF**

□ 4. **Optimizing regular path expressions using graph schemas**
Fernandez, M.; Suciu, D.;
Data Engineering, 1998. Proceedings., 14th International Conference on
23-27 Feb. 1998 Page(s):14 - 23
Digital Object Identifier 10.1109/ICDE.1998.655753
AbstractPlus | Full Text: PDF(224 KB) **IEEE CNF**

□ 5. **Parallelization and object-orientation: a database processing point of vie**
Taniar, D.; Rahayu, J.W.;
Technology of Object-Oriented Languages, 1997. TOOLS 24. Proceedings
Sept. 1997 Page(s):244 - 253
Digital Object Identifier 10.1109/TOOLS.1997.713550
AbstractPlus | Full Text: PDF(488 KB) **IEEE CNF**

□ 6. **Using interface definition languages to support path expressions and prc contract**
Watkins, D.;
Technology of Object-Oriented Languages, 1998. TOOLS 26. Proceedings

3-7 Aug. 1998 Page(s):308 - 317
Digital Object Identifier 10.1109/TOOLS.1998.711022
AbstractPlus | Full Text: PDF(188 KB) IEEE CNF

□ 7. **Visualization of path expressions in a visual object-oriented database qu**
Jeonghee Kim; Taisook Han; Suk Kyoon Lee;
Database Systems for Advanced Applications, 1999. Proceedings., 6th Interna
on
19-21 April 1999 Page(s):99 - 108
Digital Object Identifier 10.1109/DASFAA.1999.765741
AbstractPlus | Full Text: PDF(1200 KB) IEEE CNF

□ 8. **CMIS/P++: extensions to CMIS/P for increased expressiveness and efficie manipulation of management information**
Pavlou, G.; Liotta, A.; Abbi, P.; Ceri, S.;
INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer a
Communications Societies. Proceedings. IEEE
Volume 2, 29 March-2 April 1998 Page(s):430 - 438 vol.2
Digital Object Identifier 10.1109/INFCOM.1998.665059
AbstractPlus | Full Text: PDF(896 KB) IEEE CNF

□ 9. **IEEE standard Verilog hardware description language**
IEEE Std 1364-2001
2001 Page(s):0_1 - 856
AbstractPlus | Full Text: PDF(3773 KB) IEEE STD

□ 10. **Discovering structural association of semistructured data**
Ke Wang; Huiqing Liu;
Knowledge and Data Engineering, IEEE Transactions on
Volume 12, Issue 3, May-June 2000 Page(s):353 - 371
Digital Object Identifier 10.1109/69.846290
AbstractPlus | References | Full Text: PDF(516 KB) IEEE JNL

□ 11. **Representing and querying changes in semistructured data**
Chawathe, S.S.; Abiteboul, S.; Widom, J.;
Data Engineering, 1998. Proceedings., 14th International Conference on
23-27 Feb. 1998 Page(s):4 - 13
Digital Object Identifier 10.1109/ICDE.1998.655752
AbstractPlus | Full Text: PDF(136 KB) IEEE CNF

□ 12. **Logic-based query optimization for object databases**
Grant, J.; Gryz, J.; Minker, J.; Raschid, L.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 12, Issue 4, July-Aug. 2000 Page(s):529 - 547
Digital Object Identifier 10.1109/69.868906
AbstractPlus | References | Full Text: PDF(328 KB) IEEE JNL

□ 13. **Index design for structured documents based on abstraction**
Chow, J.-H.; Cheng, J.; Chang, D.; Xu, J.;
Database Systems for Advanced Applications, 1999. Proceedings., 6th Interna
Conference on
19-21 April 1999 Page(s):89 - 96
Digital Object Identifier 10.1109/DASFAA.1999.765740
AbstractPlus | Full Text: PDF(168 KB) IEEE CNF

□ 14. **Deriving path expressions recursively**
Bertolino, A.; Marre, M.;
Program Comprehension, 1993. Proceedings., IEEE Second Workshop on
8-9 July 1993 Page(s):177 - 185
Digital Object Identifier 10.1109/WPC.1993.263893
AbstractPlus | Full Text: PDF(532 KB) IEEE CNF

□ 15. **Comparison of parallel algorithms for path expression query in object da**
Guoren Wang; Ge Yu; Kaneko, K.; Makinouchi, A.;
Database Systems for Advanced Applications, 2001. Proceedings. Seventh In
Conference on
18-21 April 2001 Page(s):250 - 257
Digital Object Identifier 10.1109/DASFAA.2001.916385
AbstractPlus | Full Text: PDF(600 KB) **IEEE CNF**

□ 16. **Design and performance evaluation of parallel algorithms for path expres database systems on NOW**
Qiang Fang; Guoren Wang; Ge Yu; Kaneko, K.; Makinouchi, A.;
Database Applications in Non-Traditional Environments, 1999. (DANTE '99) Pr
International Symposium on
1999 Page(s):395 - 402
Digital Object Identifier 10.1109/DANTE.1999.844984
AbstractPlus | Full Text: PDF(316 KB) **IEEE CNF**

□ 17. **Realtime system control by means of path expressions**
Schoute, A.L.; Luursema, J.J.;
Real Time, 1990. Proceedings., Euromicro '90 Workshop on
6-8 June 1990 Page(s):79 - 86
Digital Object Identifier 10.1109/EMWRT.1990.128232
AbstractPlus | Full Text: PDF(596 KB) **IEEE CNF**

□ 18. **An augmented Web space for digital cities**
Hiramatsu, K.; Ishida, T.;
Applications and the Internet, 2001. Proceedings. 2001 Symposium on
8-12 Jan. 2001 Page(s):105 - 112
Digital Object Identifier 10.1109/SAINT.2001.905173
AbstractPlus | Full Text: PDF(776 KB) **IEEE CNF**

□ 19. **Analysis and evaluation of resources utility strategies for parallel path ex algorithms on DSVM**
Qiang Fang; Guoren Wang; Ge Yu;
High Performance Computing in the Asia-Pacific Region, 2000. Proceedings.
International Conference/Exhibition on
Volume 2, 14-17 May 2000 Page(s):862 - 864 vol.2
Digital Object Identifier 10.1109/HPC.2000.843561
AbstractPlus | Full Text: PDF(220 KB) **IEEE CNF**

□ 20. **WebView: a tool for retrieving internal structures and extracting informat documents**
Lim, S.-J.; Ng, Y.-K.;
Database Systems for Advanced Applications, 1999. Proceedings., 6th Interna
Conference on
19-21 April 1999 Page(s):71 - 80
Digital Object Identifier 10.1109/DASFAA.1999.765738
AbstractPlus | Full Text: PDF(572 KB) **IEEE CNF**

□ 21. **A synchronization mechanism for inheritance in POOSL**
Sook-Young Choi; Kwan-Jong Yoo; Hyeok-Chan Kwon;
High Performance Computing on the Information Superhighway, 1997. HPC A
28 April-2 May 1997 Page(s):708 - 711
Digital Object Identifier 10.1109/HPC.1997.592237
AbstractPlus | Full Text: PDF(356 KB) **IEEE CNF**

□ 22. **Parallel double sort-merge algorithm for object-oriented collection join q**
Taniar, D.; Rahayu, W.;
High Performance Computing on the Information Superhighway, 1997. HPC A
28 April-2 May 1997 Page(s):122 - 127
Digital Object Identifier 10.1109/HPC.1997.592134
AbstractPlus | Full Text: PDF(532 KB) **IEEE CNF**

□ 23. **A knowledge-based system for patient image pre-fetching in heterogene environments - modeling, design, and evaluation**
Chih-Ping Wei; Paul Jen-Hwa Hu; Sheng, O.R.L.;
Information Technology in Biomedicine, IEEE Transactions on
Volume 5, Issue 1, March 2001 Page(s):33 - 45
Digital Object Identifier 10.1109/4233.908382
AbstractPlus | References | Full Text: PDF(188 KB) **IEEE JNL**

□ 24. **Integrating concurrency and object-orientation using boolean, access an**
Sajeev, A.S.M.; Schmidt, H.W.;
High Performance Computing, 1996. Proceedings. 3rd International Conferenc
19-22 Dec. 1996 Page(s):68 - 72
Digital Object Identifier 10.1109/HIPC.1996.565799
AbstractPlus | Full Text: PDF(304 KB) **IEEE CNF**

□ 25. **Path expression processing in Korean natural language query interface f oriented databases**
Jinseok Chae; Sukho Lee;
Systems, Man and Cybernetics, 1995. 'Intelligent Systems for the 21st Centur International Conference on
Volume 1, 22-25 Oct. 1995 Page(s):690 - 695 vol.1
Digital Object Identifier 10.1109/ICSMC.1995.537844
AbstractPlus | Full Text: PDF(368 KB) **IEEE CNF**

[redacted]

**View: 1-25** | 26-

Help Contact Us Privacy &

Indexed by
Inspec

Google

**Web** Images Groups News Froogle Local **more »**

"a high-level debugging mechanism" bruegge | Search | Advanced Search Preferences

**Web** Results **1** - **10** of about **133** for **"a high-level debugging mechanism" bruegge**. (**0.32** seconds)

Citations: Generalized Path Expressions: **A High-Level Debugging ...**
B. **Bruegge** and P. Hibbard. Generalized Path Expressions: **A High-Level ...**
Similarly, **Bruegge** and Hibbard [BH83] use generalized path expressions that refer **...**
citeseer.ist.psu.edu/context/159846/0 - 18k - Cached - Similar pages

Related: The Design and Implementation of a Run-Time Analysis and ...
2: Generalized Path Expressions: **A High-Level Debugging Mechanism** (context) - **Bruegge**, Hibbard - 1983 2: Dalek: A GNU improved programmable debugger **...**
citeseer.ist.psu.edu/nrelatedgid/351860 - 8k - Cached - Similar pages

[PDF] Geheralized Path Expressions: **A High Level Debugging Mechanism ...**
File Format: PDF/Adobe Acrobat
**A High Level Debugging Mechanism**. Preliminary Draft. Bernd **Bruegge**. Peter Hibbard. Department of Computer Science. Carnegie Mellon University **...**
portal.acm.org/ft_gateway.cfm?id=1006157&type=pdf - Similar pages

Generalized path expressions
Generalized path expressions: **a high level debugging mechanism ...** 5 B. **Bruegge**. Debugging in a Multiple Language Environment Thesis Proposal, **...**
portal.acm.org/citation.cfm?id=1006140.1006157 - Similar pages
[ More results from portal.acm.org ]

References
4: B. **Bruegge** and P. Hibbard. Generalized path expressions: **A high level debugging mechanism**. In Proceedings of the ACM SIGSOFT/SIGPLAN Software Engineering **...**
www.cs.uoregon.edu/~sameer/ ariadne/papers/pdcs94/paper/node12.html - 8k - Cached - Similar pages

References
6: B. **Bruegge** and P. Hibbard. Generalized path expressions: **A high level debugging mechanism**. In Proceedings of the ACM SIGSOFT/SIGPLAN Software Engineering **...**
www.cs.uoregon.edu/~sameer/ ariadne/papers/journ/node12.html - 20k - Cached - Similar pages
[ More results from www.cs.uoregon.edu ]

Entry **Bruegge**:1983:GPE from sigplan1980.bib
@Article{**Bruegge**:1983:GPE, author = "B. **Bruegge** and P. Hibbard", title = "Generalized path expressions: **a high level debugging mechanism**", **...**
www.math.utah.edu:8080/ftp/pub/ tex/bib/idx/sigplan1980/18/8/34-44.html - 31k - Cached - Similar pages

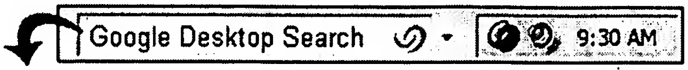
[PDF] A Dataflow Language for Scriptable Debugging
File Format: PDF/Adobe Acrobat - View as HTML
[7] B. **Bruegge** and P. Hibbard. Generalized path expressions:. **A high level debugging mechanism**. In Proceedings of. the ACM SIGSOFT/SIGPLAN Software **...**
www.cs.brown.edu/~sk/Publications/Papers/ Published/mckr-dataflow-lang-script-debug/paper.pdf - Similar pages

Citebase - A Tracer Driver for Versatile Dynamic Analyses of ...
B. **Bruegge** and P. Hibbard. Generalized path expressions:**A high-level debugging mechanism**. The Journal of Systems and Software, 3:265-276, 1983. Elsevier. **...**
citebase.eprints.org/cgi-bin/ citations?id=oai:arXiv.org:cs/0508105 - 25k - Cached - Similar pages

[PDF] "Lightweight" Semantics Models for Program Testing and Debugging ...
File Format: PDF/Adobe Acrobat - View as HTML
[9] B. **Bruegge**, P. Hibbard, "Generalized Path Expressions: **A High-Level Debugging**

**Mechanism**", The. Journal of Systems and Software 3, 1983, pp. 265-276. ...
www.disi.unige.it/person/ ReggioG/PROCEEDINGS/auguston.pdf - Similar pages

Gooooogle ▶
Result Page: 1 2 3 4 5 Next



"a high-level debugging mechanism" Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google